


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☒ The Guide


**THE GUIDE TO COMPUTING LITERATURE**

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **null\_check pei intermediate representation**

 Found **38** of **910,869**

Sort results by

Display results


[Save results to a Binder](#)

[Search Tips](#)
☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The Digital Library](#)

Results 1 - 20 of 38

 Result page: **1** [2](#) [next](#)

 Relevance scale ☐ ☐ ☐ ☐ ☐
**1** [Efficient and precise modeling of exceptions for the analysis of Java programs](#)


Jong-Deok Choi, David Grove, Michael Hind, Vivek Sarkar

 September 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1999 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering PASTE '99**, Volume 24 Issue 5

Publisher: ACM Press

Full text available: pdf(1.16 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Factored Control Flow Graph, *FCFG*, is a novel representation of a program's intraprocedural control flow, which is designed to efficiently support the analysis of programs written in languages, such as Java, that have frequently occurring operations whose execution may result in exceptional control flow. The FCFG is more compact than traditional CFG representations for exceptional control flow, yet there is no loss of precision in using the FCFG. In this paper, we introduce the FCFG r ...

**2** [Exploring the Interaction between Java's Implicitly Thrown Exceptions and Instruction Scheduling](#)


Matthew Arnold, Michael Hsiao, Ulrich Kremer, Barbara G. Ryder

 April 2001 **International Journal of Parallel Programming**, Volume 29 Issue 2

Publisher: Kluwer Academic Publishers

 Full text available: [Publisher Site](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The frequent occurrence of implicitly thrown exceptions poses one of the challenges present in a Java compiler. Not only do these implicitly thrown exceptions directly affect the performance by requiring explicit checks, they also indirectly impact the performance by restricting code movement in order to satisfy the precise exception model in Java. In particular, instruction scheduling is one transformation that is restricted by implicitly thrown exceptions due to the heavy reliance on reordering ...

**Keywords:** Java exceptions instruction scheduling

**3** [Intermediate representation engineering: Efficient online optimization by utilizing offline analysis and the safeTSA representation](#)


Jeffery von Ronne, Andreas Hartmann, Wolfram Amme, Michael Franz

 June 2002 **Proceedings of the inaugural conference on the Principles and Practice of programming, 2002 and Proceedings of the second workshop on**